JOURNAL OF COMPUTER AND SYSTEM SCIENCES: 5, 54–66 (1971)

# The Use of Lists in the Study of Undecidable Problems in Automata Theory*

J. HARTMANIS AND F. D. LEWIS†

*Department of Computer Science, Cornell University, Ithaca, N. Y. 14850*

Received January 23, 1970

Many of the problems in automata theory are unsolvable and can be classified into degrees of unsolvability by their relative difficulty. In this note, natural reference sets are presented which belong to the complete degrees at each level of the arithmetic hierarchy. Also, some questions regarding lists of recursively enumerable sets are considered. These results resolve some apparent peculiarities and provide simple methods of determining the degrees of unsolvability for several well-known problems and permit easy construction of natural problems with high degrees of unsolvability.

## 1. INTRODUCTION

It is well known that many problems in automata theory and formal languages are recursively unsolvable and, furthermore, that these problems can be classified into degrees of unsolvability by their relative difficulty. In this note we show, first, how to construct in a natural way sets of Turing machines which are complete sets in any desired level of the arithmetic degress of unsolvability. The construction of these complete reference sets mirrors directly the quantifier sequences of the arithmetic hierarchy. Next, a method of viewing difficult problems as simpler problems about lists of Turing machines will be presented.

For example, the problem whether a Turing machine accepts a cofinite set is equivalent to the problem of determining whether a recursive list of Turing machines contains a machine which accepts every input sequence. The problem whether a Turing machine accepts every input sequence is, in turn, equivalent to the problem whether each Turing machine on a recursive list accepts some input. Viewed differently, it can be seen that the cofiniteness problem is also equivalent to the problem of determining whether a recursive list of Turing machines contains only finitely many machines which accept no input.

These examples suggest that there should be some general results which describe

* This research was supported in part by National Science Foundation Grant No. GJ-155.

† Current address: Computation Laboratory, Harvard University, Cambridge, Mass. 02138.

how the difficulty of a property $P$ increases when we ask "cardinality" questions about machines with property $P$ on a recursive list of machines. Utilizing our complete reference sets we prove several such results. These results clarify some apparent anomalies in the degrees of unsolvability of problems about lists and provide an overview about such problems. They, furthermore, yield simple methods to determine the minimal degree of unsolvability of several well-known problems and permit easy construction of natural problems with high degrees of unsolvability.

## 2. Preliminaries and Reference Sets

We assume that the reader is familiar with the basic notions of Turing machines, recursive and recursively enumberable (r.e.) sets, as defined in [2, 5, 11], as well as with undecidable problems and their classification by their relative difficulty [6, 9].

For the sake of completeness we introduce some notation and review a few definitions about reducibility and the arithmetic hierarchy at this point. The reader unfamiliar with these concepts should consult the above references. Let $M_0$, $M_1$, $M_2$,... be an admissable [10] enumeration of Turing machines or partial recursive functions, and let $W_0$, $W_1$, $W_2$,... denote the sets accepted by the corresponding machines. We write $T(i, x, t) = 1$ if and only if (iff) the $i$-th Turing machine $M_i$ halts on input $x$ in $t$ steps. (This is similar to the Kleene $T$-predicate.) Finally, $\theta$ *Property* designates the set of indices (index set) for Turing machines that accept sets which have that particular "*Property*."

The set $A$ is *one-one reducible* to $B$, which is written $A \leqslant_1 B$, iff there exists a one-one recursive function $g$, such that for all $x$

$$x \in A \Leftrightarrow g(x) \in B.$$

The set $A$ is *one-one equivalent* to $B$, $A \equiv_1 B$, iff $A \leqslant_1 B$ and $B \leqslant_1 A$. These concepts were introduced by Post [8].

Next we outline the classification of undecidable problems by means of the arithmetic or Kleene hierarchy [4].

We say that a predicate $S$ is *recursive* if $S$ is recursively decidable. Furthermore, $S$ is an *arithmetic predicate* iff $S$ is obtained from some recursive predicate by means of quantification.

The set $A$ is an *arithmetic set* iff there is some arithmetic predicate $S$ such that

$$A = \{x \mid S(x)\}.$$

An arithmetic predicate $S$ can always be written in *prenex normal form* of a recursive predicate $R$ and a prefix of quantifiers [5, p. 167; 11, p. 308]. These quantifiers can be collapsed so that they alternate and then these predicates are placed in the hierarchy

according to the number of alternating existential and universal quantifiers in the prefixes.

(a) $S$ is a *$\Sigma_n$-predicate* iff its prefix begins with an existential quantifier, $\exists$, and contains $n$ alternating quantifiers.

(b) $S$ is a *$\Pi_n$-predicate* iff its prefix begins with an universal quantifier, $\forall$, and contains $n$ alternating quantifiers.

For example,

$$S(x) = \exists u\, \forall v\, \exists w\, R(u, v, w, x) \text{ is a } \Sigma_3\text{-predicate,}$$

provided that $R(u, v, w, x)$ is a recursive predicate.

A set $A$ is an *$\Sigma_n(\Pi_n)$ set* iff there exists a $\Sigma_n(\Pi_n)$-predicate $S$ such that

$$A = \{x \mid S(x)\}.$$

For the sake of brevity we will write $A \in \Sigma_n$ or $A \in \Pi_n$.

The set $A$ is *$\Sigma_n$-complete* iff $A \in \Sigma_n$ and every $\Sigma_n$ set is one-one reducible to $A$. *$\Pi_n$-complete* sets are defined similarly.

Several well-known index sets which will be used later are defined:

(a) $\theta\ Empty = \{i \mid M_i \text{ accepts no inputs}\}$
$= \{i \mid \forall\, x\, \forall t\, [T(i, x, t) \neq 1]\}$
is a $\Pi_1$-complete set.

(b) $\theta\ Finite = \{i \mid M_i \text{ accepts a finite number of inputs}\}$
$= \{i \mid \exists x\, \forall y\, \forall t[y > x \Rightarrow T(i, y, t) \neq 1]\}$
is a $\Sigma_2$-complete set.

(c) $\theta\ Every = \{i \mid M_i \text{ accepts all inputs}\}$
$= \{i \mid \forall x\, \exists t[T(i, x, t) = 1]\}$
is a $\Pi_2$-complete set.

(d) $\theta\ Cofinite = \{i \mid M_i \text{ accepts all but a finite number of inputs}\}$
$= \{i \mid \exists x\, \forall y\, \exists t[y > x \Rightarrow T(i, y, t) = 1]\}$
is a $\Sigma_3$-complete set.

To construct complete sets of higher degress is considerably more difficult and many, say $\Sigma_5$-complete, examples are cumbersome to describe. Furthermore, these examples are adhoc and the proofs that they are complete sets of a given degree may be quite difficult.

To overcome these and other difficulties we introduce complete reference sets of the arithmetic hierarchy. These sets arise naturally from the concepts of Turing machines and alternating quantifiers and will be used later to prove results about lists of machines.

DEFINITION. The *n-th acceptance set* $A_n$ is

$$\{i \mid \forall x_1 \exists x_2 \cdots \forall x_{n-1} \exists x_n [M_i \text{ accepts } x_1 \# x_2 \# \cdots \# x_{n-1} \# x_n]\}$$

when $n$ is even and

$$\{i \mid \exists x_1 \forall x_2 \cdots \forall x_{n-1} \exists x_n [M_i \text{ accepts } x_1 \# x_2 \# \cdots \# x_n]\}$$

when $n$ is odd (where the $x_i$ do not contain $\#$).

DEFINITION. The *n-th rejection set* $R_n$ is

$$\{i \mid \exists x_1 \forall x_2 \cdots \forall x_n [M_i \text{ does not accept } x_1 \# \cdots \# x_n]\}$$

when $n$ is even and

$$\{i \mid \forall x_1 \exists x_2 \cdots \forall x_n [M_i \text{ does not accept } x_1 \# \cdots \# x_n]\}$$

when $n$ is odd (again the $x_i$ do not contain $\#$).

For example, the set $A_3$ consists of all indices of Turing machines for which there is a (binary) sequence $x$ such that for every (binary) sequence $y$ there exists a sequence $z$ such that the machine accepts the input $x \# y \# z$. Clearly $A_n = \bar{R}_n$, $R_n = \bar{A}_n$, and $R_1 = \theta\ \mathit{Empty}$.

Now we shall place the reference sets in the Arithmetic hierarchy by showing that they are complete in their respective degrees of unsolvability.

THEOREM. (*Completeness*).

(a) *If n is odd then* $A_n$ *is* $\Sigma_n$*-complete and* $R_n$ *is* $\Pi_n$*-complete.*

(b) *If n is even then* $A_n$ *is* $\Pi_n$*-complete and* $R_n$ *is* $\Sigma_n$*-complete.*

*Proof.* Only part (b) will be proven for $A_n$ since the other proofs are quite similar. First, $A_n \in \Pi_n$ since $i \in A_n$ can be expressed by

$$\forall x_1 \exists x_2 \cdots \exists x_n \exists t\{T(i, x_1 \# \cdots \# x_n, t) = 1].$$

Next select any set $B \in \Pi_n$. This set can be expressed by

$$B = \{z \mid \forall x_1 \exists x_2 \cdots \exists x_n R(x_1, \ldots, x_n, z)\},$$

for some recursive predicate $R$.

Consider the machine $M_{g(z)}$ which accepts only inputs of the form $x_1 \# \cdots \# x_n$ iff $R(x_1, \ldots, x_n, z)$ is true. Therefore

$$\begin{aligned} z \in B &\Leftrightarrow \forall x_1 \exists x_2 \cdots \exists x_n R(x_1, \ldots, x_n, z) \\ &\Leftrightarrow \forall x_1 \cdots \exists x_n [x_1 \# \cdots \# x_n \in W_{g(z)}] \\ &\Leftrightarrow g(z) \in A_n. \end{aligned}$$

The function $g$ is recursive (by Church's Thesis) and one-one since a new machine $M_{g(z)}$ is constructed for each $z$.

Therefore, $B \leqslant_1 A_n$ and $A_n$ is $\Pi$-complete.

This result shows that we can locate our reference sets in each step of the arithmetic hierarchy and therefore they are equivalent to the reference sets commonly used in recursion theory. (These are $\phi^{(n)}$ and $\overline{\phi^{(n)}}$ and are presented in [11]). Hence all of the well known results such as the hierarchy theorem for $A_n$ and $R_n$. Also the standard reducability relationships $A_n \leqslant_1 A_{n+1}$, $R_n \leqslant_1 R_{n+1}$, $A_n \leqslant_1 R_{n+1}$, and $R_n \leqslant_1 A_{n+1}$ are true. These reference sets can be used to determine the minimal degree of unsolvability for automata theory problems in a natural way.

For example, $\theta$ *Every* a well known $\Pi_2$-complete set can be shown to be equivalent to $A_2$ rather easily. To see this, reduce $\theta$ *Every* to $A_2$ via the one-one, recursive function $g$ which is defined as

$$M_{g(i)} \text{ accepts } x \# y \text{ iff } T(i, x, y) = 1.$$

The reduction of $A_2$ to $\theta$ *Every* is achieved via $f$,

$$M_{f(i)} \text{ accepts } x \text{ iff } \exists y\, \exists t[T(i, x \# y, t) = 1].$$

Other applications of these reference sets appear in the following sections.

## 3. Lists

Questions about lists are common in automata theory and many problems, even if not so stated originally, can be better understood when they are expressed as problems about lists.

In this section we derive several results which, for a property $P$, characterize the degree of unsolvability of recursively invariant problems about lists of machines in terms of the set associated with $P$ and the structure of the problem.

Definition. A *list* is a recursively enumerable set. (These will be denoted $W_i$ in that they may be accepted by $M_i$).

Definition. The property $P$ is called *recursively invarient* iff it is preserved by recursive isomorphisms or one-one, onto, recursive functions. (That is, if $A$ has property $P$ and $g$ is a one-one, onto, recursive function then $g(A)$ has $P$ also.)

Note that the standard cardinality properties, "$A$ is empty," "$A$ is finite," "$A$ is infinite," and "$A$ is coinfinite" are recursively invariant.

In the following, most of the lists mentioned will consist of indices for Turing machines. The questions asked about the lists will usually be about the intersection of a list and some well-known family of machines (or index set).

The next lemma shows that in recursively invariant questions about intersection with list we can replace any complete set by its corresponding reference set.

LEMMA. *If $P$ is a recursively invariant property and $B \equiv_1 A_n$ then*

$$C = \{i \mid P(A_n \cap W_i) = 1\} \equiv_1 \{i \mid P(B \cap W_i) = 1\} = D.$$

*Proof.* (a) Let $B \leqslant_1 A_n$ via the one-one, onto recursive function $g$ and define the function $f$ such that $W_{f(i)} = g(W_i)$. (This can be done easily from $g$.) Figure 1 should help in visualizing the meaning of $g$ and $f$.

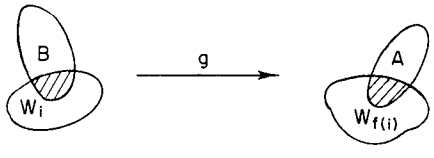


FIGURE 1

Since $g$ is one-one and onto it preserves recursively invariant properties (by a theorem of Myhill [7]). Thus

$$P(B \cap W_i) = P(g(B \cap W_i)) = P(A_n \cap W_{f(i)}).$$

And therefore

$$\begin{aligned} i \in D &\Leftrightarrow P(B \cap W_i) = 1 \\ &\Leftrightarrow P(A_n \cap W_{f(i)}) = 1 \\ &\Leftrightarrow f(i) \in C. \end{aligned}$$

So $D \leqslant_1 C$ via the function $f$.

(b) $C \leqslant_1 D$ can be shown in a similar manner.

This lemma is also true for sets equivalent to $R_n$ and is proven in exactly the same way.

The next result indicates how the difficulty of a problem increases when we ask a question about the problem's intersection with a list of Turing machine indices. Because of the previous lemma we can clearly ask the questions about $A_n$ or $R_n$ and have them carry over to any equivalent problem.

The next two theorems demonstrate two methods of proof. The first is a machine based reduction while the second is a more elegant proof based on a result of Kreisel, Schoenfield, and Wang. Both kinds of proof are given here so that the methods will be demonstrated.

THEOREM. *If $A$ is $\Sigma_n$-complete then the set $\{i \mid A \cap W_i = W_i\} = \{i \mid W_i \subseteq A\}$ is $\Pi_{n+1}$-complete.*

*Proof.* Because of the previous lemma the set $A_n$ can be used in place of $A$. Therefore let $B = \{i \mid A_n \cap W_i = W_i\}$.

(a) $B$ is a $\Pi_{n+1}$ set since

$$\begin{aligned} i \in B &\Leftrightarrow A_n \cap W_i = W_i \Leftrightarrow \forall x[x \in W_i \Rightarrow x \in A_n] \\ &\Leftrightarrow \forall x[x \notin W_i \quad \text{or} \quad x \in A_n] \\ &\Leftrightarrow \forall x[\forall t[T(i, x, t) \neq 1] \quad \text{or} \quad x \in A_n] \\ &\Leftrightarrow \forall x\, \forall t[T(i, x, t) \neq 1 \quad \text{or} \quad x \in A_n]. \end{aligned}$$

Since $A_n \in \Sigma_n$ then $B \in \Pi_{n+1}$ and therefore $B \leqslant_1 A_{n+1}$.

(b) Recalling that

$$A_n = \{i \mid \exists x_1 \cdots \exists x_n[x_1 \# \cdots \# x_n \in W_i]\}$$

and

$$A_{n+1} = \{i \mid \forall m\, \exists x_1 \cdots \exists x_n[m \# x_1 \# \cdots \# x_n \in W_i]\}.$$

For any machine $M_i$ construct the list $W_{g(i)} = \{k_0, k_1, ...\}$ of machine indices in the following manner: Machine $M_{k_m}$ of the list rejects all inputs not of the form $x_1 \# \cdots \# x_n$. When presented with some $x_1 \# \cdots \# x_n$, $M_{k_m}$ merely writes $m\#$ on front, and processes $m \# x_1 \# \cdots \# x_n$ as $M_i$ would.

The indices, $k_m$, and the list of them $W_{g(i)}$, are both easy to define in terms of $M_i$.

Intuitively, if $i \in A_{n+1}$, then all $k_m$ are elements of $A_n$. More formally,

$$\begin{aligned} i \in A_{n+1} &\Leftrightarrow \forall m\, \exists x_1 \cdots \exists x_n[m \# x_1 \# \cdots \# x_n \in W_i] \\ &\Leftrightarrow \forall m\, \exists x_1 \cdots \exists x_n[x_1 \# \cdots \# x_n \in W_{k_m}] \\ &\Leftrightarrow \forall m[k_m \in A_n] \\ &\Leftrightarrow A_n \cap W_{g(i)} = W_{g(i)} \Leftrightarrow g(i) \in B. \end{aligned}$$

Therefore $A_{n+1} \leqslant B$.

This theorem can be proven for $R_n \in \Sigma_n$ in the same manner.

Now sets in higher degrees can be constructed in a natural way from sets in lower degrees. For any $\Sigma_n$-complete set $A$, the set of lists (actually their indices) that contain only elements of $A$ is $\Pi_{n+1}$-complete. Several examples (corollaries) to the above Theorem are:

(a) The set of lists of (indices of) Turing machines which accept finite sets,

$$\{i \mid W_i \cap \theta\, Finite = W_i\}.$$

is $\Pi_3$-complete.

(b) The set of lists of Turing machines which accept cofinite sets,

$$\{i \mid W_i \cap \theta \textit{ Cofinite} = W_i\},$$

is $\Pi_4$-complete.

The above theorem cannot be generalized to $\Pi_n$-complete sets. It turns out that the set of lists which contains only elements from a $\Pi_n$-complete set $A$ is again a $\Pi_n$ set. An examination of the structure of the set

$B = \{i \mid W_i \cap R_n = W_i\}$ when $R_n \in \Pi_n$ explains this fact,

$$\begin{aligned} i \in B &\Leftrightarrow W_i \cap R_n = W_i \\ &\Leftrightarrow \forall x[x \in W_i \Rightarrow x \in R_n] \\ &\Leftrightarrow \forall x\, \forall t[T(i, x, t) \neq 1 \quad \text{or} \quad x \in R_n]. \end{aligned}$$

The last line is a $\Pi_n$ form since $x \in R_n$ is a $\Pi_n$ predicate.

Another property of lists produces sets two levels above the original set. This is accomplished by asking if a list contains an infinite number of elements from some set rather than all of the list being elements of the set.

THEOREM. *If $A$ is $\Pi_n$-complete then the set $\{i \mid A \cap W_i$ is infinite$\}$ is $\Pi_{n+2}$-complete.*

*Proof.* Again let $B = \{i \mid A \cap W_i$ is infinite$\}$.

(a) $B \in \Pi_{n+2}$ since

$$\begin{aligned} i \in B &\Leftrightarrow A \cap W_i \text{ is infinite} \\ &\Leftrightarrow \forall x\, \exists y[y > x \wedge y \in W_i \wedge y \in A] \\ &\Leftrightarrow \forall x\, \exists y\, \exists t[y > x \wedge T(i, y, t) = 1 \wedge y \in A] \\ &\Leftrightarrow \forall \exists \exists [y \in A \wedge (\text{recursive predicate})]. \end{aligned}$$

Since $y \in A$ is a $\Pi_n$ predicate, $B \in \Pi_{n+2}$.

(b) A lemma of Kreisel, Schoenfield, and Wang [6] concerning the quantifier $Ux$ which means "there exist infinitely many $x$" is used to prove this part. This result reveals that any $\Pi_{2n}$ predicate may be stated:

$$Ux_1 \cdots Ux_n\, R(x_1, \ldots, x_n, y)$$

for some recursive $R$ and for any predicate in $\Pi_{2n+1}$ the form will be

$$Ux_1 \cdots Ux_n\, \forall x_{n+1}\, R(x_1, \ldots, x_{n+1}, y).$$

Therefore let $C$ be any $\Pi_{n+2}$ set and by the above arrangement it may be represented by

$$x \in C \Leftrightarrow UyS(x, y)$$

for some $\Pi_n$-predicate $S$. Since $A$ is a $\Pi_n$-complete set, there is a recursive, one-one function $f$ such that

$$S(x, y) \Leftrightarrow f(x, y) \in A.$$

Define $M_{g(x)}(y) = f(x, y)$ and let the set $W_{h(x)}$ be the range of $M_{g(x)}$. Then

$$\begin{aligned} x \in C &\Leftrightarrow UyS(x, y) \\ &\Leftrightarrow Uy[f(x, y) \in A] \\ &\Leftrightarrow W_{h(x)} \cap A \text{ is infinite} \\ &\Leftrightarrow h(x) \in B. \end{aligned}$$

And so $C \leqslant_1 B$, making $B$ $\Pi_{n+2}$-complete.

With this result, sets can be constructed that are two levels above the original set. Several immediate examples are:

(a) The set of lists of Turing machines of which an infinite number of machines on the list accept no inputs,

$$\{i \mid W_i \cap \theta \textit{ Empty} \text{ is infinite}\}$$

is a $\Pi_3$-complete set.

(b) The set of lists of Turing machines where an infinite number on the list accept all inputs;

$$\{i \mid W_i \cap \theta \textit{ Every} \text{ is infinite}\}$$

is a $\Pi_4$-complete set.

(c) The set of lists of Turing machines where infinitely many machines on the list accept a coinfinite set;

$$\{i \mid W_i \cap \theta \textit{ Coinfinite} \text{ is infinite}\}$$

is as suspected, a $\Pi_5$-complete set.

Note that when this procedure is applied to sets which are $\Sigma_n$ complete, the new set is only of level $\Pi_{n+1}$. Again, this is explained by the structure of the membership predicate for the new set. Assuming that $A \in \Sigma_n$ and reviewing part (a) of the previous proof, we arrive at the expression:

$$\forall\exists\exists[y \in A \wedge \cdots].$$

Since $A \in \Sigma_n$, this predicate is merely a $\Pi_{n+1}$ form, hence only a single jump may be obtained.

There are many natural properties of lists (like the two mentioned above), these properties take the form of sets like

$$\{i \mid P(A, W_1) = 1\},$$

where $P(A, W_i)$ is a property such as: "Is there an element of $A$ on the list $W_i$" or "Are there a finite number of elements of $A$ on the list $W_i$." Some of the obvious properties are shown in the chart given below. The proofs for these remain as an easy application of the methods used in the previous proofs.

| $P(A, W_i)$ | $A$ is $\Sigma_n$-complete | $A$ is $\Pi_n$-complete |
|---|---|---|
| $A \cap W_i = \phi$ | $\Pi_n$-complete | $\Pi_{n+1}$-complete |
| $A \cap W_i \neq \phi$ | $\Sigma_n$-complete | $\Sigma_{n+1}$-complete |
| $A \cap W_i = W_i$ | $\Pi_{n+1}$-complete | $\Pi_n$-complete |
| $A \cap W_i \neq W_i$ | $\Sigma_{n+1}$-complete | $\Sigma_n$-complete |
| $A \cap W_i$ is finite | $\Sigma_{n+1}$-complete | $\Sigma_{n+2}$-complete |
| $A \cap W_i$ is infinite | $\Pi_{n+1}$-complete | $\Pi_{n+2}$-complete |
| $\bar{A} \cap W_i$ is finite | $\Sigma_{n+2}$-complete | $\Sigma_{n+1}$-complete |
| $\bar{A} \cap W_i$ is infinite | $\Pi_{n+2}$-complete | $\Pi_{n+1}$-complete |

## 4. Further Applications

The properties of lists indicated by the above chart simplify the reduction process required to provide a set's lower bound in the degrees of unsolvability. For instance, instead of a complicated priority arguement [11], a simple reduction from the set $\{i \mid W_i \cap \theta\ \textit{Empty}$ is finite$\}$ is all that is required to show that $\theta\ \textit{Cofinite}$ is a $\Sigma_3$-complete set.

We give two illustrations of this technique.

Theorem. *$\theta\ \textit{Finite} = \{i \mid W_i$ is finite$\}$ is $\Sigma_2$-complete.*

*Proof.* (a) Writing $\theta\ \textit{Finite}$ in prenex normal form;

$$\begin{aligned} i \in \theta\ \textit{Finite} &\Leftrightarrow \exists x\, \forall y[y > x \Rightarrow y \notin W_i] \\ &\Leftrightarrow \exists x\, \forall y[y \leqslant x \vee y \notin W_i] \\ &\Leftrightarrow \exists x\, \forall y[y \leqslant x \vee \forall t(T, i, y, t) \neq 1)] \\ &\Leftrightarrow \exists \forall \forall [\text{recursive predicate}]. \end{aligned}$$

And so $\theta\ \textit{Finite}$ is an $\Sigma_2$ and therefore $\theta\ \textit{Finite} \leqslant_1 R_2$ by the Completeness Theorem.

(b) Now $R_2 \leqslant_1 \theta\ \textit{Finite}$ must be shown. Consider the set $B = \{i \mid W_i \cap A_1$ is finite$\}$ which, from the chart on list properties, is $\Sigma_2$-complete and therefore $B \equiv R_2$. ($A_1$ is the complement of $\theta\ \textit{Empty}$.) The reduction of $B$ to $\theta\ \textit{Finite}$ proceeds as follows:

From any machine $M_i$, another machine, $M_{g(i)}$ is constructed so that whenever $M_i$ accepts a list in which only a finite number of elements are not indices for the empty set, then $M_{g(i)}$ accepts a finite set. More formally, let

$$W_i = \{a_0, a_1, \cdots\} \text{ and consider}$$

$$M_{g(i)}(k) = \begin{cases} \text{halt if } \exists a_k \in W_i \text{ and } \exists x[M_{a_k}(x) \text{ halts}] \\ \text{diverge otherwise.} \end{cases}$$

Therefore whenever $a_k$ is an index for the empty set (or when $W_i$ is finite and there is no $a_k$), then $M_{g(i)}(k)$ does not halt. However, when $a_k$ exists and is a member of $A_1$ then $M_{g(i)}(k)$ halts. Thus

$$\begin{aligned} i \in B &\Leftrightarrow W_i \cap A_1 \text{ is finite} \\ &\Leftrightarrow \exists \text{ finite } k \text{ for which } M_{g(i)}(k) \text{ halts} \\ &\Leftrightarrow g(i) \in \theta \textit{ Finite}. \end{aligned}$$

That is, since $A_1$ contains all of the indices for the nonempty sets, if $W_i \cap A_1$ is finite then all but a finite number of elements of $W_i$ are indices for the empty set. Therefore $M_{g(i)}$ diverges for all but a finite number of inputs.

THEOREM. *$\theta$ Cofinite is $\Sigma_3$-complete.*

*Proof.* (a) First the prenex normal form,

$$\begin{aligned} i \in \theta \textit{ Coinfinite} &\Leftrightarrow \exists x \, \forall y[y > x \Rightarrow y \in W_i] \\ &\Leftrightarrow \exists x \, \forall y[y \leqslant x \vee y \in W_i] \\ &\Leftrightarrow \exists x \, \forall y[y \leqslant x \vee \exists t(T(i, y, t) = 1)] \\ &\Leftrightarrow \exists \forall \exists [\text{recursive predicate}], \end{aligned}$$

and therefore $\theta$ *Cofinite* $\in \Sigma_3$ and $\theta$ *Cofinite* $\leqslant_1 A_3$.

(b) Next the reduction of the set

$B = \{i \mid W_i \cap \theta$ *Empty* is finite$\}$ (which is $\Sigma_3$-complete) to $\theta$ *Cofinite.*

Taking any $M_i$, let $W_i = \{a_0, a_1, ...\}$ as usual and define the machine $M_{g(i)}$:

$$M_{g(i)}(k) = \begin{cases} \text{halt if } \exists x[M_{a_k}(x) \text{ halts}], \\ \text{diverge otherwise.} \end{cases}$$

If $a_k$ is the index for a null set then $M_{g(i)}(k)$ diverges, and if only a finite number of the $a_i$ are in $\theta$ *Empty* then $M_{g(i)}$ diverges for a finite number of inputs.

Or,

$$i \in B \Leftrightarrow W_i \cap \theta\ \textit{Empty} \text{ is finite}$$

$$\Leftrightarrow W_{a_k} = \phi \text{ for a finite number of } k$$

$$\Leftrightarrow k \notin W_{g(i)} \text{ for a finite number of } k$$

$$\Leftrightarrow g(i) \in \theta\ \textit{Cofinite}$$

Thus $B \leqslant_1 \theta$ *Cofinite* and $\theta$ *Cofinite* is $\Sigma_3$-complete.

The above proofs are all rather straightforward, and the same construction was used in each one. By careful selection of the set $B$, this is the case with most proofs of this type.

The approach and techniques developed in this note have various applications in automata theory. We conclude by illustrating how they can be used to solve problems about recursively enumerable lists of context-free (c.f.) grammars. Problems of this type have been recently studied by Cudia[1].

Through the use of techniques developed in [3] we can easily show that:

(a) the set of ambiguous context-free grammars is $\Sigma_1$-complete;

(b) the set of context-free grammars which generate regular languages (or cofinite sets) is $\Sigma_2$-complete.

Thus, we immediately conclude that the set of all recursively enumerable lists which contain c.f. grammars which generate nonregular sets is $\Sigma_3$-complete.

The set of r.e. lists of c.f. grammars in which each list contains infinitely many grammars generating regular sets is $\Pi_3$-complete. On the other hand, the set of lists of c.f. grammars in which each list contains infinitely many grammars which generate non-regular sets is $\Pi_4$-complete.

The set of r.e. lists of c.f. grammars which contain only a finite number of ambiguous grammars is $\Sigma_2$-complete, on the other hand, the set of lists of c.f. grammars which contains infinitely many unambiguous grammars is $\Pi_3$-complete.

Most of the results about Turing equivalence of undecidable problems about lists of context-free grammars drop out from considerations of this type and the apparent anomolies are easily explained by the sequences of quantifiers describing the desired sets.

## Acknowledgment

The authors wish to express their gratitude to an anonymous referee for providing the elegant proof to the second theorem on lists.

## REFERENCES

1. D. F. CUDIA, The degree hierarchy of undecidable problems of formal grammars, *in* "2nd Ann. ACM. Symp. on Theory of Comput.," pp. 10–21, ACM, New York, 1970.
2. M. DAVIS, "Computability and Unsolvability," McGraw-Hill, New York, 1958.
3. J. HARTMANIS, Context free languages and Turing machine computations, *Proc. Symp. Appl. Math.* **19** (1967), 42–51.
4. S. C. KLEENE, Recursive predicates and quantifiers, *Trans. Amer. Math. Soc.* **53** (1943), 41–73.
5. S. C. KLEENE, "Introduction to Metamathematics," Van Nostrand, Princeton, N. J., 1952.
6. G. KREISEL, J. R. SHOENFIELD, AND H. WANG, Number theoretic concepts and recursive well-orderings, *Arch. Math. Logik Grundlagenforsh* **5** (1960), 42–64.
7. J. MYHILL, Creative sets, *Z. Math. Logik Grundlagen Math.* **1** (1955), 97–108.
8. E. L. POST, Recursively enumerable sets of positive integers and their decision problems, *Bull. Amer. Math. Soc.* **50** (1944), 284–316.
9. H. ROGERS, JR., Computing degrees of unsolvability, *Math. Ann.* **138** (1959), 125–140.
10. H. ROGERS, JR., Gödel numbering of partial recursive functions, *J. Symbolic Logic* **23** (1958), 331–341.
11. H. ROGERS, JR., "Theory of Recursive Functions and Effective Computability," McGraw-Hill, New York, 1967.